Rethinking CPU Foundations

Do we need registers?

Dr Chris Crispin-Bailey
Department of Computer Science
University of York

MIDAS Consortia
Context

• **Towards Zero Power Computing (ZPC)**
  – we need leaner processor cores

But also …

• **High Performance and Low power**
  - Data to New Knowledge – demanding on power
Zero Powered ICT
– not a single scenario – many variations

Many low/zero power nodes
E.G Dispersed data acquisition

Mixed capabilities
low-powered nodes with High perf.
Zero Powered ICT → Data to new Knowledge

Scalable Data Processing nodes with High performance

Multicore & scalability
Data to New Knowledge
But power is a key concern
Motivation

• So low power and high performance is desirable

• Is current CPU design thinking optimal?
  - almost certainly the answer is no but it *is* convenient
  - 'big' corporations have invested too much to change..

• Are there alternatives?
  – what are they?
  - is anyone considering this? - MIDAS
Example

- Vast majority of CPU’s are dominated by register files

BUT this immediately has down-stream design impact …

- Multi-porting of register files is costly (power/area/delay)

- Causes RAW/WAR hazards
  - *May demand Reorder buffers, register renaming, - costly*

- Impact on pipeline architecture and delay behaviour
  - *need increasingly complex schemes to hide pipeline penalties*

- Scales poorly with silicon technology progression.
Example

• **Implicit Data-flow Execution Architectures**
  - Superscalar Stack based coding
  - No addressable register file - no RAW/WAR hazards
  - No need for ROB/renaming logic, etc
  - Higher cache density – or smaller cache – reduced external I/O

- Unique storage element properties (will come to this)

- **But is IDEA ‘better’ than REGISTER?**
  - unknown
  - research is needed on this and other novel paradigms
IDEA – unique storage behaviour

Reachable die area in 16Fo4 delay metric

We can measure useable chip area using a delay metric of $16.F04^*$

This translates into an absolute delay for a chosen process and therefore frequency of Operation

$F04 = \text{unit delay of one standard inverter}$

In any given process technology
For a Register file, all operands emerge at the same time, leaving a uniform reachable chip area for related components.

Due to register file delay, the reachable area is already reduced considerably.
But with IDEA paradigm, operands are available in skewed time.

This could be exploited in many ways

- Asynchronicity
- more floor-planning flexibility
- Multiplexing of operand buses
  - e.g. 8 operands of 32 bits = 256 wires
    this is costly for both area and power
  - IDEA - 4 time separated pairs = 64 wires
Rethinking CPU Foundations - conclusion

Difficult to cover in depth in 10 minutes – one example is given

But hopefully we see that :-

• CHIST-ERA covers many levels
• Low power and high performance must be part of the solution
• Register file may be easiest, but not necessarily the best

Therefore, we need to question current CPU fundamentals, and look seriously at alternatives.

The CHIST-ERA programme may be a platform to do this …
MIDAS project proposal consortia

University of York - Chris Crispin-Bailey
University of Amsterdam – Chris Jesshope, Raphael Kena
University of Limerick – Brendan Mullane

Multicore Implicit Dataflow Architectures and Systems

Aim: - to build an 'IDEA' core prototype chip
- tool-chain - programmability
- component library – building blocks & methodology
- real silicon
- evaluate as a multicore component
Thankyou.

Dr Chris Crispin-Bailey

chrisb@cs.york.ac.uk

www.cs.york.ac.uk/~chrisb